อาจหาญ สัตยารักษ์

SOFT MAIL

ทางเลือกใหม่ในการจัดการ ฐานข้อมูลดีเบส (4)

ชุดคำสั่งปาสคาลเพิ่มเติมในการพิมพ์ข้อมูลดีเบส

ในการจัดการกับฐานข้อมูลดีเบสด้วยปาสคาลในตอนที่ผ่านมา ผู้เขียนหวังว่าท่านคงพบแนวทางใหม่ที่ช่วยให้มีทางออกในการจัดการกับฐานข้อมูลดีเบสได้อีกทางหนึ่ง คราวนี้เราต่อตอนที่ 4 โดยจะเป็นการพิมพ์ข้อมูลที่เราต้องการลงกระดาษ ซึ่งผู้เขียนเลือกใช้การพิมพ์อักขระแบบกราฟิกเพราะจะไม่มีปัญหาในเรื่องการทำรหัสภาษาไทยที่เครื่องพิมพ์แล้วหรือยัง รายละเอียดการพิมพ์อักขระแบบกราฟิกนี้อยู่ในวารสารฉบับที่ 63-64 ผู้ที่ประสงค์จะทำความเข้าใจก็ให้ย้อนไปอ่านได้

โปรแกรมสำหรับตอนนี้อยู่ในไฟล์ชื่อ dbase4.pas โดยมีฐานข้อมูลดีเบสตัวอย่างชื่อ dbase4.dbf ซึ่งมีโครงสร้างฐานข้อมูลดังนี้

no(N,4), name(C,25), address(C,25), district(C,20), province(C,15) และ postcode(C,5)

นอกจากนี้ยังมีแฟ้มเอกสาร 2 แฟ้ม dbase4-1, dbase4-2 มียูนิตไฟล์ชื่อ font9.tpu และ font24.tpu สำหรับการพิมพ์ข้อมูลลงกระดาษโดยเป็นการพิมพ์แบบกราฟิก โดยทั้งสองยูนิตนี้ผู้เขียนทำมาจากบทความเรื่องการพิมพ์อักขระแบบกราฟิกในฉบับที่ 63-64 ตามลำดับ เป็นการคอมพายล์ด้วยปาสคาล 5.5 ถ้าผู้อ่านใช้ปาสคาลรุ่นอื่นก็ให้ทำยูนิตไฟล์ทั้งสองขึ้นลง โดยต้องมีชุดคำสั่งที่เรียกใช้งานได้คือ copybyte และ print

เปลี่ยนแปลงการกำหนดตัวแปร

เพื่อให้เราสามารถเรียกข้อมูลจากฐานดีเบสให้มากที่สุดเท่าที่ปาสคาลจะยอมได้ จึงขอเปลี่ยนการอ่านข้อมูลจากของเดิมมาเป็นการเรียกเข้าสู่หน่วยความจำที่เรียกว่า "ฮีพ" แทน ซึ่งจะทำให้อ่านข้อมูลได้ครั้งละ 65535 ไบต์ ในชื่อตัวแปร DataDbase ส่วนตัวแปร WordDbase กำหนดไว้รองรับข้อมูลจากแฟ้มเอกสารซึ่งจะใช้งานร่วมกับฐานข้อมูลดีเบสอีกที

เริ่มด้วยการเรียกใช้ยูนิต crt,dos, Font24 (ถ้าเป็นเครื่องพิมพ์ 9 เข็มให้เรียก Font9 แทน)

พร้อมกันนี้ก็ได้กำหนดตัวแปรสตริงชื่อ WordST เป็นตัวแปรมิติซึ่งใช้หน่วยจำฮีพเช่นกัน เพื่อเก็บข้อความที่อ่านได้จากแฟ้มเอกสาร

ตัวแปร NoF ใช้สำหรับเก็บหมายเลขเขตข้อมูลที่จะให้พิมพ์ลงกระดาษ

ตัวแปร P1, P2, SP สำหรับเก็บค่าต่าง ๆ ที่จำเป็นต่อการพิมพ์ข้อมูล

รายละเอียดดูได้จากรูปที่ 1

จากการที่เราเปลี่ยนแปลงตัวแปรที่จะรองรับข้อมูลไปจากเดิม จึงทำให้การอ่านข้อมูลต้องเปลี่ยนไปด้วย ดังนั้นชุดคำสั่ง read_DataBase ต้องเปลี่ยนมาใช้คำสั่งตามรูปที่ 2 ซึ่งได้แก่คำสั่ง blockread นอกนั้นจะคงแบบเดิม

เก็บข้อมูลในเขตข้อมูล

จากรูปที่ 2 นั้นจะเห็นชุดคำสั่งชื่อ DataInFiled ซึ่งจะใช้ในการนำเอาข้อมูลในระเบียนข้อมูลที่กำหนด (R) มาบรรจุไว้ในตัวแปร FieldSTR[No] ซึ่งเป็นตัวแปรมิติ โดยจะมีการทำงานดังนี้

กำหนดค่าเริ่มต้น xx=1

ตรวจตั้งแต่เขตข้อมูลที่ 1 ถึงเขตข้อมูลสุดท้าย (CountOfField)

หาจุดเริ่มต้นของระเบียนข้อมูล คือ ค่า x

นำข้อมูลตามขอบเขตความยาวของเขตข้อมูลมาใส่ในตัวแปร Field STR[No]

เพิ่มค่า xx ตามความยาวของเขตข้อมูล เพื่อหาข้อมูลบนเขตข้อมูลถัดไป

เราจะได้ข้อมูลในเขตข้อมูลต่าง ๆ มาเมื่อจบการทำงานของชุดคำสั่งนี้ ซึ่งจะนำไปใช้ในการพิมพ์ลงกระดาษต่อไป

การพิมพ์จำหน้าซองจดหมาย

การพิมพ์ข้อมูลจากฐานข้อมูลดีเบสลงกระดาษนั้นมีหลายรูปแบบ แต่ผู้เขียนจะเสนอเพียง 2 รูปแบบ นั่นคือ การพิมพ์ครั้งละ 1 ระเบียนข้อมูลโดยพิมพ์

```
PROGRAM ReadDATA_from_dBASE_IV;
uses crt,dos,Font24;
type  HeapData=^WordData;
      WordData=record
                 WordF:array[0..65534] of char;
               end;
      HeapSTR=^String;
var f:file;
    NoR,CountOfRecord,SizeOfRecord,SizeOfHead,EditNo:word;
    Head:array[0..1024] of byte;
    NameOfField:array[1..30] of string[10];
    TypeOfField:array[1..30] of char;
    WidthOfField:array[1..30] of byte;
    FDec:array[1..30] of byte;
    ch,Funckey:char;  CountOfField,AppendckNoField:byte;
    Dbasefile,DateStr,DbaseSTR,InputSTR,WordP:string;
    Numeric,Edit,UseDbf:boolean;
    FieldSTR:array[1..30] of string;  Space:string[40];
    DbaseNum,NumField:real;
    DataDbase,WordDbase:HeapData;
    Xstart,LineNo,LineMax,MaxF:word;
    WordST:array[1..120] of HeapSTR;
    NoF,P1,P2:array[1..120] of byte;
    SP:array[0..10] of byte;
```

รูปที่ 1 การกำหนดตัวแปรเดิมและตัวแปรที่เปลี่ยนแปลงใหม่สำหรับใช้งาน

```
PROCEDURE Read_DataBase;
var no:word; k:byte;
begin
   assign(f,Dbasefile);
   {$I-}  reset(f,1);
 if IOResult<>0 then  begin  UseDbf:=false;
     gotoxy(20,21); write('FILE :',DbaseFile,' not found ');
     readchr;          end
 else     Begin
   blockread(f,head,10);  SizeOfHead:=Head[8]+Head[9]*256;
   reset(f,1);            blockread(f,head,SizeOfHead);
   CountOfRecord:=Head[4]+Head[5]*256;
   SizeOfRecord:=Head[10]+Head[11]*256;
   blockread(f,DataDbase^,SizeOfRecord*CountOfRecord);
   close(f);              CountOfField:=round(SizeOfHead/32)-1;
 for no:=1 to CountOfField  do
  begin
     NameOfField[no]:=copy(Space,1,10);  Fdec[no]:=0;
     TypeOfField[no]:=chr(Head[32*no+11]);
     WidthOfField[no]:=Head[32*no+16];
     k:=0;
      repeat
         NameOfField[no][k+1]:=chr(Head[32*no+k]);
         inc(k);   if head[32*no+k]=0 then k:=27;
      until k>9;
        if TypeOfField[no]='N' then  FDec[No]:=Head[32*no+17];
   end;     end;
end;


PROCEDURE DataInField(R:word)
var no,s,x,xx:word;
begin
   xx:=1;
 for No:=1 to CountOfField do
   begin
     if R<=CountOfRecord then
       begin  FieldSTR[No]:='';
        x:=(R-1)*SizeOfRecord+xx;
        for s:=x to (x+WidthOfField[No]-1) do
                 FieldSTR[No]:=FieldSTR[No]+DataDbase^.WordP[s];
       end;
      xx:=xx+WidthOfField[no];
   end;
end;
```

รูปที่ 2 ชุดคำสั่งอ่านข้อมูลจากดีเบสโดยอาศัยหน่วยความจำฮีพ

บรรทัดละหนึ่งเขตข้อมูลตามเขตข้อมูลที่ต้องการ กับการพิมพ์บรรทัดละ 1 ระเบียนข้อมูล คือ มีหลายเขตข้อมูลอยู่ในบรรทัดเดียวกัน ซึ่งแต่ละแบบนั้นได้เสนอมา 2 วิธี คือ แบบกำหนดให้พิมพ์โดยตรงเลย กับแบบที่กำหนดรูปแบบการแสดงผลโดยอาศัยโปรแกรมพิมพ์เอกสารทั่ว ๆ ไปมาช่วย

หากท่านต้องการเป็นอย่างอื่นก็ให้ลองเขียนชุดคำสั่งขึ้นมาโดยอาศัยแนวคิดที่เสนอไว้นี้

เมื่อเรามีฐานข้อมูลดีเบสที่เก็บรายชื่อและที่อยู่ของลูกค้า หากประสงค์จะนำข้อมูลนั้น ๆ มาจ่าหน้าซองจดหมายก็สามารถทำได้ดังชุดคำสั่งในรูปที่ 3

แฟ้มข้อมูลที่เรียกมาใช้งานคือ dbase4.dbf ซึ่งมีเขตข้อมูลดังนี้

no(N,4), name(C,25), address (C,25), district(C,20), province(C,15) และ postcode(C,5)

การพิมพ์เราต้องการนำเขตข้อมูล name, address, district, province และ postcode ซึ่งเป็นเขตข้อมูลลำดับที่ 2,3, 4,5 และ 6 ตามลำดับมาใช้งาน

จากชุดคำสั่ง PrintOut_I มีรายละเอียดดังนี้

ตัวแปร dotout เป็นตัวแปรที่กำหนดในยูนิต font24.tpu หรือ font 9.tpu เพื่อกำหนดระยะห่างระหว่างบรรทัดของการพิมพ์ในที่นี้กำหนดเป็น #5 ถ้ากำหนดมาขึ้นก็จะมีระยะระหว่างบรรทัดห่างกันมากขึ้น

อ่านไฟล์ฐานข้อมูลดีเบสชื่อ dbase 4.dbf เข้ามาสู่หน่วยความจำ

กำหนดค่าเริ่มต้นของหมายเลขระเบียนข้อมูลที่จะพิมพ์ที่ no:=Rst; ถ้าค่านี้น้อยกว่าจำนวนระเบียนสูงสุดก็ให้ทำงานวนรอบ repeat

```
PROCEDURE Printout_I(Rst,Rend:word);
var No:word;

begin  dotOut:=#5;   CursorOff;
   Dbasefile:='dbase4.dbf';  Read_DataBase;  No:=RSt;
 if No<CountOfRecord then begin
   repeat
       DataInField(NO);
       print('     ');      print(' ');
       print('           '+FieldSTR[2]);
       print('           '+FieldSTR[3]);
       print('           '+FieldSTR[4]);
       print('           '+FieldSTR[5]);
       print('           '#5+FieldSTR[6]+#5);
       Copybyte(#10#10#10);
       Inc(No);  if no>CountOfRecord then No:=Rend+1;
  until no>Rend;           end;
    CursorOn;
end;
```

รูปที่ 3 ชุดคำสั่งเพื่อพิมพ์จ่าหน้าซองจดหมายจากฐานข้อมูลดีเบสแบบที่ 1

นำข้อมูลในระเบียนข้อมูลที่กำหนด (ระเบียนที่ no) มาใส่ในตัวแปร FieldSTR[] ด้วยการเรียกใช้ชุดคำสั่ง DataInField

ทำการพิมพ์ข้อความว่า เรียน และบรรทัดว่างหนึ่งบรรทัด

พิมพ์ข้อมูลที่อยู่ในเขตข้อมูลลำดับที่ 2 คือ name

พิมพ์ข้อมูลที่อยู่ในเขตข้อมูลลำดับที่ 3 คือ address

พิมพ์ข้อมูลที่อยู่ในเขตข้อมูลลำดับที่ 4 คือ district

พิมพ์ข้อมูลที่อยู่ในเขตข้อมูลลำดับที่ 5 คือ province

พิมพ์ข้อมูลที่อยู่ในเขตข้อมูลลำดับที่ 6 คือ postcode โปรดสังเกตว่าส่วนมีรหัส ^PE(#5) ควบคุมอยู่ซึ่งจะทำให้ได้ผลการพิมพ์ของเขตข้อมูลนี้เป็นตัวขยาย

ตามด้วยเลื่อนกระดาษ 1 บรรทัด หากต้องการให้เลื่อนทั้งหน้าให้ใช้คำสั่ง copybyte(#12) แทน

ซึ่งเป็นการจ่าหน้าซองหนึ่งซองต่อหนึ่งระเบียนข้อมูล จากนั้นทำการเลื่อนไปยังเขตข้อมูลลำดับถัดไป แล้วย้อนไปพิมพ์แบบเดิมกับระเบียนข้อมูลใหม่จนกว่าจะจบระเบียนข้อมูลที่ต้องการ

การกำหนดรูปแบบด้วยแฟ้มเอกสาร

วิธีการในการกำหนดรูปแบบของการจ่าหน้าซองจดหมายอีกวิธีหนึ่ง คือ การกำหนดรูปแบบของการพิมพ์ด้วยโปรแกรมพิมพ์เอกสารโดยทั่ว ๆ ไป แล้วให้เขียนโปรแกรมภาษาปาสคาลขึ้นมาอ่าน โดยมีหลักการดังนี้

1. กำหนดชื่อเขตข้อมูลด้วยอักขระ ! ปิดหัวท้าย เช่น !name! ก็หมายถึงเขตข้อมูล name หรือ !postcode! ก็คือเขตข้อมูล postcode เป็นต้น ส่วนอื่น ๆ ที่ไม่ได้กำหนดเป็นเขตข้อมูลก็จะเป็นการพิมพ์แบบปกติ โดยกำหนดให้บรรทัดแรกเป็นชื่อไฟล์ฐานข้อมูลดีเบส ข้อความที่จะพิมพ์อยู่ตั้งแต่บรรทัดที่ 2 เป็นต้นไป

2. อ่านแฟ้มเอกสารเข้ามาสู่หน่วยความจำ พร้อมกับนำไปเก็บไว้ในตัวแปรชื่อ WordST[] เรียงลำดับจากตัวแปร

มิติลำดับที่ 1 ไปจนถึงตัวแปรมิติลำดับที่ LineMax ซึ่งคือจำนวนบรรทัดสูงสุดของแฟ้มเอกสาร โดยการตรวจจุดสิ้นสุดของตัวแปรละตัวที่รหัส #10 และจบที่รหัส #26 ดังชุดคำสั่ง ReadWord ในรูปที่ 4 ซึ่งขอให้ผู้อ่านศึกษาการทำงานของชุดคำสั่งนี้เอง เพราะคิดว่าคงเข้าใจไม่ยาก โดยขอให้สังเกตการใช้ตัวแปรในฮีพว่ามีรูปแบบอย่างไรด้วย

3. นำข้อความที่บรรจุในตัวแปร WordST[] มาตรวจหาลำดับของเขตข้อมูลที่ต้องการพิมพ์ ตามชุดคำสั่ง FindNoField_I ซึ่งจะได้ค่าตัวแปร NoF ลำดับของเขตข้อมูล และ P1,P2 บอกตำแหน่งของรหัส ! ที่ใช้เป็นเครื่องชี้ว่าเป็นชื่อเขตข้อมูล โดยทั้งสองตัวแปรนี้มีไว้นำข้อมูลในเขตข้อมูลที่ระบุเข้ามาแทนที่ส่วนนี้ของแฟ้มเอกสาร อันทำให้ได้ผลการพิมพ์ตรงกับความต้องการ

4. เมื่อทราบลำดับของเขตข้อมูล ตำแหน่งการแทนที่ข้อมูลแล้วก็มาถึงการนำข้อมูลจากเขตข้อมูลนั้นไปใส่ในตัวแปร WordST[] ทำได้ตามชุดคำสั่ง setFieldToWord ซึ่งขอให้สังเกตการใช้

รูปที่ 4 ชุดคำสั่งในการอ่านแฟ้มเอกสารที่ใช้งานร่วมกับฐานข้อมูลดีเบส

```
PROCEDURE ReadWord(Filename:string);
var size:word;
begin
  assign(f,Filename); reset(f,1);
  size:=filesize(f);
  blockread(f,WorddBase^,size);
  close(f);
  LineNo:=1; Size:=0;
  repeat
   WordST[LineNo]^:='';
   repeat
    if WorddBase^.WordF[size] in [#1..#9,#11,#12,#14..#25,#28..#255] then
           WordST[LineNo]^:=WordST[LineNo]^+WorddBase^.WordF[size];
      inc(Size);
   until WorddBase^.WordF[size] in [#10,#26];
     LineMax:=LineNo; Inc(LineNo);
  until WorddBase^.WordF[size]=#26;
end;

PROCEDURE FindNoField_I;
var x:word; P,No:byte;     STT:string;
begin
  for x:=2 to LineMax do
      begin
        STT:=WordST[x]^;    p:=pos('!',STT);
        if p<>0 then           begin
              STT:=Copy(STT,P+1,length(STT)-P);  P1[x]:=P;
              p:=pos('!',STT);  P2[x]:=P+P1[x];
           if p<>0 then
              begin
                STT:=CUT(Copy(STT,1,P-1));
                for p:=1 to length(STT) do STT[p]:=Upcase(STT[p]);
                No:=1; repeat
                         if STT=copy(NameOfField[no],1,length(STT)) then
                                                  NoF[x]:=No;
                       Inc(No);
                      until (NoF[x]<>0) or (No>CountOfField);
              end;             end;
        end;
end;

PROCEDURE setFieldTOWord(N:byte);
begin
  WordP:=copy(WordST[n]^,1,p1[N]-1)+FieldSTR[NOF[n]]+
         copy(WordST[n]^,p2[n]+1,length(WordST[n]^)-p2[N]);
  Print(WordP);
end;
```

ตัวแปร NoF, P1 และ P2 เมื่อได้ข้อมูลแล้วก็ให้พิมพ์ลงกระดาษด้วยคำสั่ง print

ในตัวอย่างเรียกใช้โปรแกรมพิมพ์เอกสารชื่อ dbase4-1 มากำหนดรูปแบบในการพิมพ์ โดยใช้มีชุดคำสั่งดังรูปที่ 5 โดยมีรายละเอียดดังนี้

กำหนดระยะห่างของบรรทัดเป็น 5 ตามด้วยการอ่านแฟ้มเอกสารเข้ามา ซึ่งจะได้ชื่อไฟล์ฐานข้อมูลดีเบสจาก DbaseFile:=cut(wordST[1]^); นำไปเปิดแฟ้มข้อมูลดีเบส

ทำการหาเขตข้อมูลที่ต้องการพิมพ์

ทำการพิมพ์ข้อมูลครั้งละหนึ่งระเบียนข้อมูล ตามคำสั่งในช่วง repeat โดยเริ่มจากหาข้อมูลที่อยู่ในเขตข้อมูลของระเบียนข้อมูลที่กำลังใช้งานจาก DataInField(no)

ในการพิมพ์จะตรวจที่ตัวแปร No F[] ว่าเป็นการพิมพ์ข้อมูล (ค่า<>0) หรือพิมพ์ข้อความตามปกติ (ค่า=0)

โดยจะจบการทำงานเมื่อค่าตัวแปรอยู่นอกขอบเขตที่กำหนด

ตามโปรแกรมนี้ผู้อ่านอาจจะกำหนดหมายเลขระเบียนเริ่มต้นและหยุด โดยใช้บรรทัดที่สองของแฟ้มเอกสารก็ได้

ผลการทำงานจะปรากฏในรูปที่ 6 โดยส่วนแรกเป็นตัวอย่างของแฟ้มเอกสารที่กำหนด ส่วนที่สองคือผลจากการพิมพ์จากชุดคำสั่งในรูปที่ 5 นั่นเอง

```
PROCEDURE Printout_II(Rst,Rend:word);
var No,NoW:word;  k:byte;

begin  dotOut:=#5;   CursorOff;
   ReadWord('dbase4-1');      Dbasefile:=cut(WordST[1]^);
   Read_DataBase;
   for NoW:=1 to 120 do NoF[NoW]:=0;
      FindNoField_I;  No:=RSt;
 if No<CountOfRecord then begin
   repeat
        DataInField(NO);
        for K:=2 to LineMax do
          begin
             if NoF[k]<>0 then  setFieldTOWord(k)
               else Print(WordST[k]^);
          end;
       Inc(No); if no>CountOfRecord then No:=Rend+1;
    until No>Rend;           end;
     CursorOn;
end;
```

รูปที่ 5 ชุดคำสั่งเพื่อพิมพ์จำหน้าซองจดหมายจากฐานข้อมูลดีเบสแบบที่ 2

```
dbase4.dbf
   เรียน

        !name!
        !address!
        !district!
        !province!
        !postcode!


    จาก วารสารไมโครคอมพิวเตอร์  กรุงเทพฯ 10400


เรียน

    นายอาจหาญ สัตยารักษ์
    โรงเรียนลำปางกัลยาณี
    อำเภอเมือง
    จังหวัดลำปาง
    52000


จาก วารสารไมโครคอมพิวเตอร์  กรุงเทพฯ 10400
```

รูปที่ 6 ตัวอย่างที่พิมพ์ได้จากชุดคำสั่งในรูปที่ 4 และ 5

การพิมพ์บรรทัดละหนึ่งระเบียนข้อมูล

ต่อไปเรามาดูแนวคิดที่สอง คือการนำข้อมูลมาพิมพ์ครั้งละหนึ่งระเบียนข้อมูลโดยพิมพ์ข้อมูลทั้งหมดในหนึ่งบรรทัด คล้ายกับการใช้คำสั่ง list แบบกำหนดชื่อเขตข้อมูล ซึ่งในการนี้เราจำเป็นต้องคำนวณหาจำนวนช่องว่างที่เหมาะสมในการที่จะพิมพ์เขตข้อมูลในต่อเนื่องในบรรทัดเดียวกัน

เพื่อความสะดวกผู้เขียนจึงกำหนดเป็นฟังก์ชันแทน เพราะจะเรียกใช้ได้สะดวกกว่าการกำหนดเป็นชุดคำสั่ง นั่น

คือฟังก์ชันชื่อ SpSTR ในรูปที่ 7 ซึ่งจะคำนวณหาจำนวนช่องว่างที่พิมพ์ต่อท้าย โดยมีความกว้างของเขตข้อมูลมาเป็นตัวกำหนด โดยตรวจข้อมูลในเขตข้อมูลนั้น ๆ ว่ามีอักขระในแถวกลาง (พวก ก, ข ฯลฯ) อยู่จำนวนกี่ตัว ตามโปรแกรมนี้ผู้เขียนใช้รหัสเกษตร ถ้าใช้รหัสอื่น ๆ ก็ต้องหาการนับจำนวนอักขระกลุ่มนี้เอาเองอีกที

โดยคำสั่งพิมพ์ก็คือชุดคำสั่งชื่อ PrintOut_III ซึ่งอยู่ในรูปที่ 7 เช่นกัน โดยในชุดคำสั่งนี้คิดว่าคงจะทำความเข้าใจได้ไม่ยาก จึงขอข้ามไม่อธิบายรายละเอียดของชุดคำสั่งนี้

ในกรณีของรูปที่ 7 เรากำหนดโดยตรงในชุดคำสั่งเลยว่าต้องการให้แสดงข้อมูลใด ต่อไปเราจะมาลองกำหนดรูปแบบในแฟ้มเอกสารขึ้นมาแทนว่าจะสามารถทำได้หรือไม่อย่างไร

เราจะเริ่มด้วยการตรวจสอบว่ามีการกำหนดให้เขตข้อมูลใดถูกนำมาพิมพ์ลงกระดาษ เนื่องจากเรากำหนดให้พิมพ์ลงในบรรทัดเดียว จึงต้องวนรอบตรวจหาอักขระ ! ในบรรทัดนั้น ๆ จนกว่าจะหาไม่เจอ โดยจะมีรายละเอียดของชุดคำสั่งดังรูปที่ 8 ชื่อ FindField_II

การตรวจสอบคราวนี้จะมีตัวแปรที่ถูกกำหนดร่วมด้วยคือ ตัวแปรมิติ SP[] ซึ่งจะเก็บตำแหน่งสดมภ์ของอักขระ ! ไว้เพื่อจะนำข้อมูลในเขตข้อมูลนั้นมาพิมพ์ ณ ตำแหน่งถัดจากตำแหน่งของอักขระ ! ไปทางขวาหนึ่งสดมภ์ ตามตัวอย่างนี้จะต้องไม่มีการเขียนข้อความภาษาไทยอื่นใดในบรรทัดที่กำหนดรูปแบบการพิมพ์นี้ ดังนั้นหากท่านประสงค์จะใส่ข้อความอื่น ๆ ด้วย ก็ต้องเขียนชุดคำสั่งตรวจอักขระภาษาไทยด้วยจึงจะได้ตำแหน่งที่ถูกต้อง

```
FUNCTION SpSTR(Content:string; B:byte):String;
 var x,BB:byte;
begin
    BB:=0;  content:=Cut(Content);
   for x:=1 to length(Content) do
    if content[x] in [#32..#214] then Inc(BB);
     SpSTR:=copy(Space,1,B-BB);
end;

PROCEDURE Printout_III(Rst,Rend:word);
var No:word;

begin  dotOut:=#5;   CursorOff;
   Dbasefile:='dbase4.dbf';  Read_DataBase;  No:=RSt;
 if No<CountOfRecord then begin
   repeat
       DataInField(NO);
       print('     '+CUT(FieldSTR[2])+SpSTR(FieldSTR[2],25)+
                     CUT(FieldSTR[3])+SpSTR(FieldSTR[3],25)+
                     CUT(FieldSTR[4])+SpSTR(FieldSTR[4],20)+
                     CUT(FieldSTR[5])+SpSTR(FieldSTR[5],15)+
                     CUT(FieldSTR[6]));
       Inc(No);  if no>CountOfRecord then No:=Rend+1;
   until no>Rend;          end;
     Print(' ');
    CursorOn;
end;
```

รูปที่ 7 ชุดคำสั่งเพื่อพิมพ์ข้อมูลจากฐานข้อมูลดีเบสแบบที่ 3

โดยถ้าตรวจพบอักขระ ! เป็นครั้งที่ 2 ก็จะตรวจหาว่าเป็นเขตข้อมูลใด (ตัวแปร NoF[x])

การพิมพ์ลงกระดาษจะเป็นไปตามชุดคำสั่ง PrintOut_IV ซึ่งขออธิบายดังนี้

อ่านแฟ้มเอกสารชื่อ dbase4-2 เข้ามา เพื่อตรวจหาชื่อไฟล์ฐานข้อมูลดีเบส (อยู่ในบรรทัดแรกของแฟ้มเอกสาร) นำไปอ่านข้อมูลเข้ามา แล้วตรวจหาเขตข้อมูลที่กำหนดในแฟ้มเอกสาร (บรรทัดที่ 3 ของแฟ้มเอกสาร) โดยจะได้ตัวแปร MaxF เป็นจำนวนของเขตข้อมูลที่จะนำมาพิมพ์

ในช่วงถัดมาของชุดคำสั่ง คือการหาหมายเลขระเบียนเริ่มต้นและสิ้นสุด ซึ่งกำหนดไว้ที่บรรทัดที่ 2 ของแฟ้มเอกสาร โดยการใช้เครื่องหมาย , เป็นตัวคั่น

รูปที่ 8 ชุดคำสั่งเพื่อพิมพ์ข้อมูลจากฐานข้อมูลดีเบสแบบที่ 4

```
PROCEDURE FindNoField_II;
var x:word; P,k,no,Fck:byte;
    STT,ST:string;
begin
  x:=0;  Fck:=0; ST:=WordST[3]^; P:=0;
for k:=1 to length(ST) do
 begin
   if ST[k]='!' then  begin  Inc(Fck);
      if Fck mod 2=1 then                    begin
        SP[x]:=k-p; P:=k; Inc(x); MaxF:=x; end
      else
        begin   STT:=CUT(copy(ST,p+1,k-p-1));
                for No:=1 to length(STT) do STT[no]:=Upcase(STT[no]);
         No:=1; repeat
                 if STT=copy(NameOfField[no],1,length(STT)) then NoF[x]:=No;
                 Inc(No);
               until (NoF[x]<>0) or (No>CountOfField);
                 writeln;
        end;            end;
 end;
end;

PROCEDURE Printout_IV;
var No,NoW:word;  k:byte; STT:string;
    Rend:word;    ck:integer;
begin  dotOut:=#5;   CursorOff;
   ReadWord('dbase4-2');      Dbasefile:=cut(WordST[1]^);
   Read_DataBase;
   for NoW:=1 to 120 do NoF[NoW]:=0;
   FindNoField_II;
   STT:=CUT(WordST[2]^); k:=pos(',',STT);
   val(copy(STT,1,k-1),No,ck); STT:=copy(STT,k+1,length(STT)-k);
   val(copy(STT,1,k-1),Rend,ck);
 if No<CountOfRecord then begin
   repeat
       DataInField(NO); STT:=copy(Space,1,SP[0]);
     for k:=1 to MaxF-1 do
         STT:=STT+CUT(FieldSTR[NoF[k]])+SpSTR(FieldSTR[NoF[k]],SP[k]);
         STT:=STT+CUT(FieldSTR[NoF[MaxF]]);   print(STT);
        Inc(No);  if no>CountOfRecord then No:=Rend+1;
    until No>Rend;            end;
      Copybyte(#27#51#30#10#10#10);
      CursorOn;
end;
```

รูปที่ 9 ตัวอย่างที่พิมพ์ได้จากชุดคำสั่งในรูปที่ 8

```
dbase4.dbf
1.3
 !name!                !address!              !district!        !province!    !postcode!


 นายอาจหาญ สัตยารักษ์    โรงเรียนลำปางกัลยาณี    อำเภอเมือง        จังหวัดลำปาง    52000
 น.ส.ชมพูนุช  สุดสวาท    23 ถนนมหาดไทยบำรุง     อำเภอเมือง        จังหวัดขอนแก่น   40000
 นายประยูร วิตา         89/7 หมู่ 7 ถนนท่าสี่ล้อ    อำเภอเมือง        จังหวัดชลบุรี    20000
```

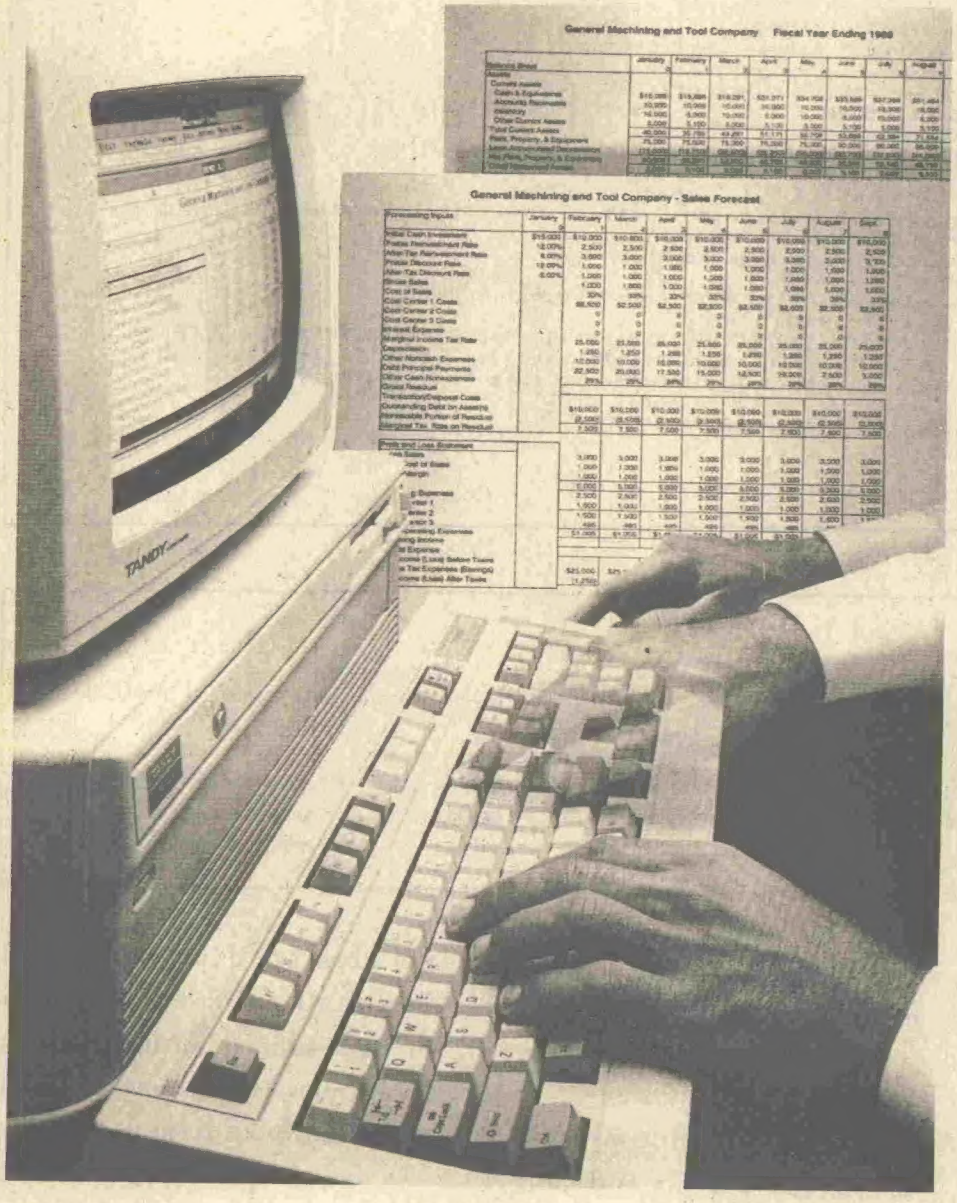
ในการพิมพ์ข้อมูลเราจะเริ่มด้วยกำหนดตัวแปร STT ให้ว่าง แล้วทำการวนรอบเพื่อนำข้อมูลที่อยู่ในเขตข้อมูลที่กำหนดของแต่ละระเบียนข้อมูลที่ใช้งานให้เข้ามาในตัวแปร STT นี้ โดยเราจะกำหนดช่องว่างต่อท้ายข้อมูลในเขตข้อมูลนั้น ๆ ด้วย เมื่อเรียบร้อยแล้วก็สั่งให้พิมพ์ด้วยคำสั่ง print(STT); ซึ่งโดยวิธีนี้จะได้ผลการพิมพ์ของแต่ละเขตข้อมูลนั้นมีแนวตรงกันเสมอ

ขอให้ดูผลการพิมพ์จากรูปที่ 9 โดยรูปส่วนบนนั้น คือ แฟ้มเอกสารชื่อ dbase4-2 ส่วนภาพล่างเป็นผลที่พิมพ์ได้จากข้อมูลในไฟล์ฐานข้อมูลดีเบสชื่อ dbase4.dbf ขอให้สังเกตว่าตำแหน่งการพิมพ์ข้อมูลจะตรงกับตำแหน่งที่กำหนดไว้ในแฟ้มเอกสารทุกประการ

รูปที่ 10 โปรแกรมหลักในการเลือกใช้การพิมพ์ทั้ง 4 แบบ

```
{======================== PrintOut dBase   ============================}

begin
   clrscr; Fillchar(Space,41,#32); Space[0]:=#40;
   for NoR:=1 to 120 do
    begin   Mark(WordST[NoR]); new(WordST[NoR]);
    end;
      Mark(DatadBase); new(DatadBase);
      Mark(WorddBase); new(WorddBase);
      Printout_I(1,3);
      Printout_II(1,3);
      Printout_III(1,3);
      Printout_IV;
   release(DatadBase); release(WorddBase);
   for NoR:=1 to 120 do release(WordST[NoR]);
end.
```



โปรแกรมหลัก

โปรแกรมในรูปที่ 10 ก็คงเป็
โปรแกรมเรียกใช้งาน โดยคราวนี้ผู้เขีย
ไปได้กำหนดเป็นแบบเมนูให้ ขอให้ผู้อ่า
ลองพัฒนาดูเองด้วย เพื่อที่จะเข้าใจกา
ทำงานต่างได้มากขึ้น

คำว่าคงทำความเข้าใจเองได้

ขอให้สังเกตการกำหนดเนื้อที่ไ
หน่วยความจำสำหรับการนำข้อมูล
กำหนดเก็บในฮีพมาใส่ ในคำสั่ง 3 คำสั่
คือ mark, new และ release ซึ่งจำเป็
ต้องกำหนดให้ตรงกับชื่อที่เรากำหนดด้ว

หากมีการกำหนดรูปแบบการพิมพ์
เป็นอย่างอื่นต่างจากนี้ก็จะทำได้โดยกา
ลองพัฒนาคำสั่งขึ้นมาใช้งานตามแนวคิด
ที่ได้อธิบายมาในตอนนี้ จึงหวังว่าค
ช่วยให้ท่านพิมพ์ข้อมูลได้รูปแบบตรงกับ
ความต้องการมากยิ่งขึ้น

พบกันใหม่ในตอนที่ 5 ครับ...